



Stellen Sie die Hochverfügbarkeit von Oracle Database 26ai mit Google Cloud NetApp Volumes bereit

NetApp database solutions

NetApp
June 12, 2026

Inhalt

Stellen Sie die Hochverfügbarkeit von Oracle Database 26ai mit Google Cloud NetApp Volumes bereit	1
Bevor Sie beginnen	1
Beispielkonfiguration, die in diesem Leitfaden verwendet wird	2
Ziele	2
Bereitstellungsoptionen	2
Abschnitte, die Sie für Ihre Stufe lesen sollten	3
Überblick	4
Architektur	4
Referenzdiagramm	4
Plattformrollen	5
Topologie und Storage	5
Compute Engine bereitstellen	6
Schritt 1: VMs erstellen	6
Schritt 2: VPC-Firewall – TCP/1521 in allen drei Zonen auf die Allowlist setzen	6
Schritt 3: Hostname, DNS und <code>/etc/hosts</code>	7
Schritt 4: Betriebssystem-Baseline (nur DB-Hosts)	8
Schritt 5: Erfassen des iSCSI-Initiatornamens (IQN)	9
GCNV iSCSI-Volumes bereitstellen	9
Schritt 1: Erstellen Sie einen GCNV Flex Unified iSCSI-Speicherpool pro Datenbankzone	9
Schritt 2: Erstellen Sie die Hostgruppen (eine pro DB-Host)	10
Schritt 3: Erstellen der GCNV iSCSI-Volumes (pro Datenbank-Host)	10
Schritt 4: Linux iSCSI und Multipath für GCNV iSCSI-Volumes konfigurieren	11
Schritt 5: Partitionieren der ASM-Backing-Geräte auf GCNV iSCSI-Volumes	13
Schritt 6: Formatieren und Einbinden <code>/u01</code> auf dem lokalen GCNV iSCSI-Volume	14
Installieren Sie die Oracle-Software	15
Schritt 1: Installieren Sie Oracle Grid Infrastructure (Oracle Restart) auf jedem DB-Host	15
Schritt 2: Installieren Sie Oracle Database 26ai auf jedem DB-Host	18
Erstellen Sie die primäre Datenbank (<code>oracdb1</code> nur)	20
Erstellen Sie die Standby-Datenbank	22
Schritt 1: Primär: SYS-Passwort, Passwortdatei und DG-Parameter	22
Schritt 2: Standby: Minimale <code>init.ora</code> pfile, Passwortdatei, NOMOUNT	23
Schritt 3: GCNV-Standby-Initialisierung	25
Schritt 4: Standby-Wiederherstellungsprotokolldateien	28
Schritt 5: Flashback auf dem Standby-System aktivieren und verwaltete Wiederherstellung starten	28
Schritt 6: Wiederherstellungsprotokoll-Versand auf dem primären Server aktivieren	29
Schritt 7: Überprüfen Sie den Ziel-Data-Guard-Status	29
Schritt 8: Registrieren Sie die Standby-Datenbank bei Oracle Restart	30
Konfigurieren Sie Data Guard Broker, FSFO und Observer	32
Schritt 1: Aktivieren Sie den Broker auf beiden Datenbanken	32
Schritt 2: Flashback bestätigen (erforderlich für die automatische Wiederherstellung durch FSFO)	33
Schritt 3: FSFO-Eigenschaften konfigurieren und aktivieren	34
Schritt 4: Installieren Sie Oracle Instant Client auf dem Observer-Host	34

Schritt 5: Starten Sie den Observer als systemd-Einheit.....	35
Schritt 6: FSFO testen (Umschaltung und Failover)	38

Stellen Sie die Hochverfügbarkeit von Oracle Database 26ai mit Google Cloud NetApp Volumes bereit

Dieser Leitfaden zeigt, wie man Recheninstanzen und Speicher bereitstellt, Oracle Grid Infrastructure und Oracle Database installiert, eine Standby-Datenbank initialisiert und Oracle Data Guard mit Fast-Start Failover konfiguriert.

Bevor Sie beginnen

Bevor Sie beginnen, stellen Sie sicher, dass Sie Folgendes haben:

- Ein Google Cloud-Projekt mit Berechtigungen für Compute Engine, VPC-Netzwerk, Firewall-Konfiguration, IAM und NetApp Volumes

Aufgabe	Erforderlicher Zugriff
Compute Engine-VMs erstellen	Compute Instance Admin (oder gleichwertig)
Firewall / Firewall-Richtlinien	Netzwerkadministrator oder delegierter Richtlinienadministrator
GCVN-Pools und -Volumes erstellen	NetApp Volumes Admin
PSA konfigurieren	Netzwerkadministrator im Host-Projekt
SSH über IAP	IAP-gesicherter Tunnelbenutzer + OS-Anmeldung (falls verwendet)

- Die NetApp Volumes-API aktiviert
- Eine für die Zielregion konfigurierte VPC und ein Subnetz
- Private Services Access (PSA) konfiguriert für Google Cloud NetApp Volumes
- Oracle Linux 10 für alle erforderlichen virtuellen Maschinen
- DNS und Hostnamensauflösung sind für die Datenbank-Hosts und den Observer-Host konfiguriert
- Oracle-Installationsmedien und Patch-Dateien für Oracle Database 26ai und Grid Infrastructure verfügbar
- Kenntnisse von Oracle Data Guard, Oracle Restart und iSCSI-Speicherkonzepten
- Die Zeitsynchronisierung ist für alle virtuellen Maschinen konfiguriert.

Sie können die folgenden Befehle verwenden:

```
gcloud services enable netapp.googleapis.com
chronyc tracking
timedatectl
```

Beispielkonfiguration, die in diesem Leitfaden verwendet wird

Dieser Leitfaden basiert auf folgenden Bereitstellungsannahmen:

- Drei virtuelle Maschinen der Google Compute Engine:
 - `oracdb1` für die primäre Datenbank
 - `oracdb2` für die Standby-Datenbank
 - `oradg-obs` für den Fast-Start Failover Observer
- Ein GCNV Flex Unified-Speicherpool pro Datenbankzone
- Fünf GCNV iSCSI-Volumes pro Datenbankhost
- Oracle Data Guard Broker und Fast-Start Failover für automatisches Failover
- Dedizierter Speicher pro Datenbank-Host; primäre und Standby-Hosts teilen sich keine iSCSI-Volumes

Ersetzen Sie alle Beispielwerte in den Befehlen durch Werte aus Ihrer Umgebung, einschließlich Hostnamen, IP-Adressen, Zonen, Projektnamen, Portal-IPs, Passwörter und Oracle-Mediendateinamen.

Ziele

In diesem Verfahren führen Sie die folgenden Aufgaben aus:

- Stellen Sie Compute Engine-Instanzen für die primäre Datenbank, die Standby-Datenbank und den Observer bereit
- Konfigurieren von GCNV iSCSI-Speicher- und Multipath-Geräten für Oracle
- Installieren Sie Oracle Grid Infrastructure und Oracle Database 26ai auf beiden Datenbankhosts.
- Erstellen Sie die primäre Oracle-Datenbank
- Initialisieren Sie die Standby-Datenbank mithilfe von GCNV-Replikation, Snapshots oder Klonen
- Konfigurieren Sie Oracle Data Guard Broker, Fast-Start Failover und den Observer

Bereitstellungsoptionen

Dieser Abschnitt vergleicht praktische HA/DR-Bereitstellungsmuster für Oracle Database auf Google Compute Engine (GCE) mit Google Cloud NetApp Volumes (GCNV) iSCSI-Blockspeicher. Außerdem wird erläutert, wie die GCNV-Replikation die Standby-Initialisierung beschleunigt. Wählen Sie anhand dieser Matrix vor Beginn die passende Ebene aus. Diese Anleitung implementiert Prod HA (Data Guard + FSFO) — die unterste Zeile.

Umfeld	Erfordernis	Empfohlene Architektur	Hochverfügbarkeit	DR	Automatisierung	Hauptvorteil
Entwicklung/Test	Niedrigste Kosten	Einzelinstanz	Nein	Ja	Nein	Snapshot-Klon
Prod Basic (Neustart)	Ausfallzeiten durch Abstürze reduzieren	+ Oracle Restart	Nein	Ja	Nur lokal	Automatischer Neustart

Umfeld	Erfordernis	Empfohlene Architektur	Hochverfügbarkeit	DR	Automatisierung	Hauptvorteil
Prod HA (kein DG)	Manuelle DR akzeptabel	+ Snapshots / RMAN	Teilweise	Ja	Teilweise	GCNV-Klonwiederherstellung
Prod HA (DG + FSFO)	Echte HA (kein DBA)	Data Guard + FSFO	Ja	Ja	Voll	Echte Hochverfügbarkeit + schnelles Failover

HA / DR / Automatisierung: Ja = erfüllt das Ziel der Stufe; Nein = nicht im Geltungsbereich; Teilweise = nur Schritte auf Speicherebene oder manuelle Schritte.

Abschnitte, die Sie für Ihre Stufe lesen sollten

Je nachdem, welches Hochverfügbarkeitsniveau Sie anstreben, lesen Sie die Abschnitte in der folgenden Matrix. Wenn Sie beispielsweise Prod HA mit Data Guard und FSFO benötigen, lesen Sie alle Abschnitte. Wenn Sie Dev/Test oder Prod Basic mit Restart benötigen, lesen Sie nur die erste Spalte.

Dev/Test oder Prod Basic (Neustart)	Prod HA (ohne Data Guard)	Prod HA (Data Guard + FSFO)
Bevor Sie beginnen	Bevor Sie beginnen	Bevor Sie beginnen
Beispielkonfiguration, die in diesem Leitfaden verwendet wird	Beispielkonfiguration, die in diesem Leitfaden verwendet wird	Beispielkonfiguration, die in diesem Leitfaden verwendet wird
Ziele	Ziele	Ziele
Bereitstellungsoptionen	Bereitstellungsoptionen	Bereitstellungsoptionen
Überblick	Überblick	Überblick
Architektur	Architektur	Architektur
Compute Engine bereitstellen	Compute Engine bereitstellen	Compute Engine bereitstellen
GCNV iSCSI-Volumes bereitstellen	GCNV iSCSI-Volumes bereitstellen	GCNV iSCSI-Volumes bereitstellen
Installieren Sie die Oracle-Software	Installieren Sie die Oracle-Software	Installieren Sie die Oracle-Software
Erstellen Sie die primäre Datenbank	Erstellen Sie die primäre Datenbank	Erstellen Sie die primäre Datenbank
	Erstellen Sie die Standby-Datenbank (nur über Schritt 3: GCNV-Standby-Initialisierung)	Erstellen Sie die Standby-Datenbank
		Konfigurieren Sie Data Guard Broker, FSFO und Observer

Überblick

Diese Architektur implementiert Oracle Database 26ai mit hoher Verfügbarkeit in der Google Cloud mithilfe von Google Cloud NetApp Volumes (GCNV) iSCSI-Speicher und Oracle Data Guard. GCNV bietet leistungsstarken Blockspeicher und unterstützt die Standby-Initialisierung auf der Speicherebene. Data Guard gewährleistet kontinuierliche Synchronisierung, Switchover und Fast-Start-Failover.

Schicht	Rolle
GCNV	Blockspeicherung, Standby-Initialisierung, Speicherreplikation
Data Guard	Kontinuierliche Synchronisierung, Redo-Anwendung, Umschaltung, FSFO

Die GCNV-Replikation verkürzt die Initialisierungszeit des Standby-Systems im Vergleich zur aktiven RMAN-Duplikation erheblich, indem Daten auf der Speicherebene anstatt über Oracle-Datenbankkanäle verschoben werden. Bevorzugen Sie die GCNV-Replikation für die Bereitstellung des Standby-Systems in der Produktion, sofern Ihre Umgebung dies unterstützt.

Komponente	Detail
DB-Hosts	oracdb1/ oracdb2 — verschiedene Zonen, jeweils fünf GCNV iSCSI-Volumes
Storage	/u01 + ASM +DATA, +RECO, +FRA auf GCNV (EXTERN)
Standby-Initialisierung	GCNV replicate/snapshot/clone → Oracle finalisieren → Data Guard
Hochverfügbarkeit	Neustart, physischer Standby (MOUNTED), Broker, FSFO, Observer (oradg-obs)
Apps	Service orclapp

Zonierung: Ein GCNV Flex Unified pool pro Datenbankzone; dedizierte Volumes pro Host. Boot-Disks sind nur für das Betriebssystem. **Nicht enthalten:** TDE, RAC, NVMe/TCP, OEM, Active Data Guard (Standby bleibt MOUNTED).

Architektur

Referenzdiagramm

Die Architektur bietet drei unabhängige Datenpfade. Speicherreplikation und Data Guard Redo-Transport sind separate Pfade.

Oracle 26ai HA on GCNV - three independent data paths

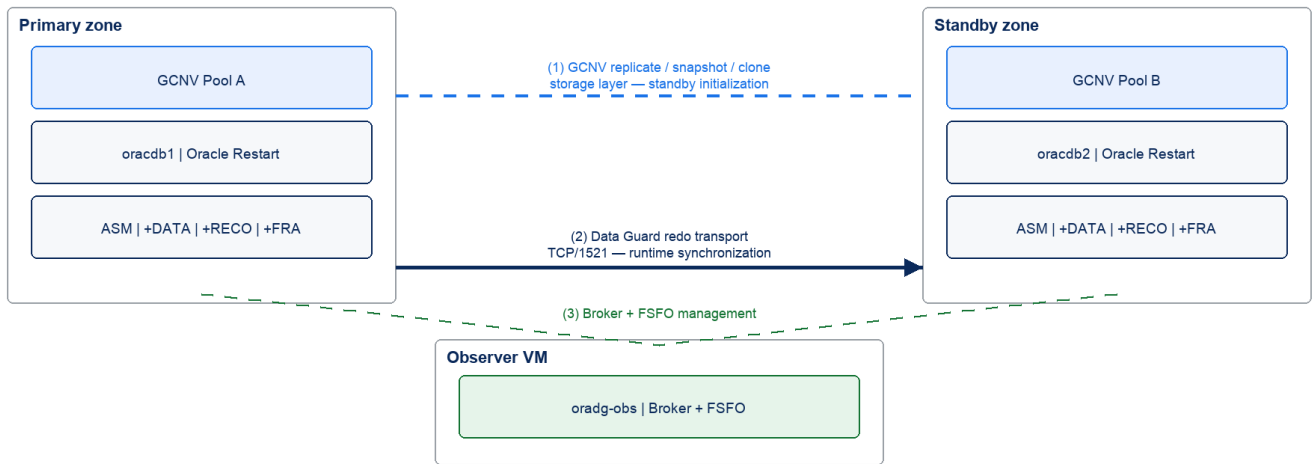


Abbildung 1. GCNV Oracle-Bereitstellung – Referenzarchitektur

Speicherreplikation ≠ Redo-Replikation. Pfad 1 verschiebt Datendateien auf der GCNV-Ebene zur Standby-Initialisierung. Pfad 2 hält die Datenbanken nach der Umstellung synchronisiert. Pfad 3 orchestriert Rollenübergänge und automatisches Failover über Oracle Data Guard Broker und den Observer.

Plattformrollen

Plattform	Liefert
GCNV	iSCSI-Volumes, Snapshots, Volume-Replikation (Baseline + Inkrementals) — Standby-Initialisierung
Data Guard	Redo-Transport, MRP, Broker, FSFO – kontinuierliche Synchronisierung und Failover

Die GCNV-Replikation führt zunächst eine Basiskopie durch und aktualisiert anschließend die Blöcke inkrementell gemäß der jeweiligen Richtlinie. Data Guard ist nach der Initialisierung des Standby für die Redo-Anwendung und FSFO zuständig.

Topologie und Storage

Rolle	VM	Zone	GCNV-Pool	Volumes (pro Host)
Primär	oracdb1	us-west1-a	oracle-pool-a	ora_<host>_u01, ora_<host>_data_01/02, ora_<host>_arch_01, ora_<host>_fra_01
Stehen zu	oracdb2	us-west1-b	oracle-pool-b	Gleiches Namensmuster

Rolle	VM	Zone	GCNV-Pool	Volumes (pro Host)
Beobachter	oradg-obs	us-west1-c	—	Nur Bootdisk

Storage	Unterstützung	Verwenden
Betriebssystem	GCE-Bootdisk	Nur Betriebssystem
/u01	GCNV iSCSI	Grid/DB-Homes, Staging (XFS)
+DATA / +RECO / +FRA	GCNV iSCSI	ASM EXTERNAL — Datendateien, Archive, FRA

Maschinentypen

Beispiel `n4-highmem-8` in diesem Leitfaden; OLTP-Workloads verwenden typischerweise `c4-standard-*`.

Compute Engine bereitstellen

Schritt 1: VMs erstellen

Erstellen Sie drei VMs in verschiedenen Zonen derselben Region (zonale Ausfallsisolierung). Bevorzugen Sie eine Region mit geringeren CO₂-Emissionen für TCO und Nachhaltigkeit, sofern sie die Anforderungen an Latenz und Compliance erfüllt (zum Beispiel `us-west1` vs `us-central1`). Verwenden Sie die Cloud Console, `gcloud`, Terraform oder Ihren Standard-Bereitstellungs-Workflow.

VM	Zone	Maschinentyp	Bootdisk	Netzwerk	Zweck
oracdb1	us-west1-a	n4-highmem-8 (Beispiel) oder c4-standard-*	OL 10, 50 GB Hyperdisk Balanced (nur Betriebssystem)	oracle-vpc / oracle-subnet, gVNIC	Primäre Datenbank
oracdb2	us-west1-b	Gleich wie primär	OL 10, 50 GB Hyperdisk Balanced (nur Betriebssystem)	Dasselbe	Standby-Datenbank
oradg-obs	us-west1-c	e2-medium	OL 10, 20 GB Hyperdisk Balanced	Dasselbe	FSFO Observer (nur Instant Client)



Netzwerkstufe: Premium, wenn Latenz oder ausgehender Datenverkehr (>~200 GiB/Monat) wichtig sind; Standard für niedrigere Gesamtbetriebskosten in Entwicklung/Test.

Aktivieren Sie **Secure Boot**, **vTPM** und **Integrity Monitoring** auf allen drei. Die Boot-Disk enthält nur das Betriebssystem. `/u01` Grid/DB-Homes, Staging und alle ASM-Daten verwenden GCNV iSCSI-Volumes (siehe [GCNV iSCSI-Volumes bereitstellen](#)) — schließen Sie **keine** separate GCE-Daten-Disk für `/u01` an.

Schritt 2: VPC-Firewall – TCP/1521 in allen drei Zonen auf die Allowlist setzen

Erlauben Sie TCP/1521 zwischen allen drei internen VM-IPs in jeder Zone `/32(us-west1-a/b/c hier)`. Verwenden Sie VPC-Firewallregeln oder Firewall Policies mit derselben Allowlist. Fehlende Regeln unterbrechen den Redo-Transport und die Observer-Konnektivität.

Cloud Console: VPC-Netzwerk → Firewall → Regel erstellen allow-oracle-net-dbhosts auf oracle-vmc — Eingehend, Zulassen, Quellen = drei /32 IPs, TCP 1521. Ausgehenden Datenverkehr spiegeln, falls erforderlich. Von jeder VM validieren:

```
sudo dnf install -y nmap-ncat

for tgt in <oracdb1-ip> <oracdb2-ip> <oradg-obs-ip>; do
  nc -zv -w 5 "$tgt" 22
  nc -zv -w 5 "$tgt" 1521
done
```

Port	Erwartet	Bedeutung
22	Verbunden	SSH-Pfad funktioniert
1521	Verbindung abgelehnt	Firewall geöffnet; Grid-Listener startet während Schritt 1: Installieren Sie Oracle Grid Infrastructure (Oracle Restart) auf jedem DB-Host
Entweder	Zeitüberschreitung	Firewall- oder Routing-Probleme beheben

Führe den Befehl von allen drei VMs zu jeder Peer-IP aus.

Schritt 3: Hostname, DNS und /etc/hosts

Nach dem Start jeder VM muss der Hostname festgelegt und /etc/hosts Einträge auf allen drei Hosts hinzugefügt werden, damit die Vorwärts-/Rückwärts-Namensauflösung für Oracle Net, den Broker und den Observer funktioniert.

```
# Run on each VM, substituting the local short name (oracdb1, oracdb2,
oradg-obs)
sudo hostnamectl set-hostname <this-host>.example.internal

# Run on every VM (same content)
sudo tee -a /etc/hosts >/dev/null <<EOF

# Oracle DG peers + FSFO Observer
<oracdb1-ip>    oracdb1.example.internal    oracdb1
<oracdb2-ip>    oracdb2.example.internal    oracdb2
<oradg-obs-ip>  oradg-obs.example.internal    oradg-obs
EOF
```

Ersetzen Sie die GCE-internen IP-Adressen (sichtbar in der Liste **Compute Engine** → **VM-Instanzen**, Spalte *Interne IP*).

Validierung von jedem Host aus:

```
ping -c 1 oracdb1 && ping -c 1 oracdb2 && ping -c 1 oradg-obs
```

Schritt 4: Betriebssystem-Baseline (nur DB-Hosts)



Voraussetzung: Ausgehendes HTTPS zu `yum.oracle.com` (Cloud NAT oder internem Mirror auf privaten Subnetzen).

Ausführen auf `oracdb1` und `oracdb2` (die Einrichtung des Beobachters wird in [Schritt 4: Installieren Sie Oracle Instant Client auf dem Observer-Host](#) behandelt).

```
# Oracle 26ai preinstall (package name varies by repo)
sudo dnf install -y oracle-ai-database-preinstall-26ai \
  || sudo dnf install -y oracle-database-preinstall-26ai \
  || sudo dnf install -y oracle-database-preinstall-23ai

# grid user + asm groups
sudo groupadd -g 54327 asmadmin; sudo groupadd -g 54328 asmdba; sudo
groupadd -g 54329 asmoper
sudo useradd -u 54322 -g oinstall -G dba,oper,asmadmin,asmdba,asmoper grid
sudo passwd -l grid; sudo passwd -l oracle
sudo usermod -a -G asmdba,asmadmin oracle

# iSCSI, multipath, OUI JDK
sudo dnf install -y iscsi-initiator-utils device-mapper-multipath
sg3_utils \
  java-21-openjdk-headless libxcrypt-compat

# THP and time
cat /sys/kernel/mm/transparent_hugepage/enabled # expect [never]
timedatectl
chronyc tracking
```



Sicherheitsstatus (OL 10): Die folgenden Befehle setzen SELinux auf den permissiven Modus und deaktivieren `firewalld`. Dies ist lediglich eine minimale Testumgebung. Für eine gehärtete SELinux- und Firewall-Konfiguration konsultieren Sie die Sicherheitsrichtlinien Ihrer Organisation.

```

sudo setenforce 0
sudo sed -i 's/^SELINUX=.*SELINUX=permissive/' /etc/selinux/config
sudo systemctl disable --now firewalld

sudo cp -n /etc/iscsi/iscsid.conf /etc/iscsi/iscsid.conf.orig
sudo sed -i '/^[#[:space:]]*node\.session\.timeo\.replacement_timeout/d'
/etc/iscsi/iscsid.conf
echo "node.session.timeo.replacement_timeout = 120" | sudo tee -a
/etc/iscsi/iscsid.conf
sudo systemctl enable --now iscsid

sudo reboot

```

Schritt 5: Erfassen des iSCSI-Initiatornamens (IQN)

Erfassen Sie nach dem Neustart den IQN jedes DB-Hosts. Sie benötigen diese IQNs, um die GCNV-Hostgruppen in [Schritt 2: Erstellen der Hostgruppen](#) zu erstellen.

```

sudo cat /etc/iscsi/initiatorname.iscsi
# InitiatorName=iqn.1994-05.com.redhat:abc123def456

```

Wiederholen Sie auf `oracdb2`. Notieren Sie die IQN jedes Hosts – eine Hostgruppe pro Host, damit ein Neustart oder eine IQN-Regenerierung eines einzelnen Hosts die GCNV iSCSI-Volumen-Sichtbarkeit eines anderen Hosts nicht beeinträchtigen kann.



Geklonte VMs: Wenn beide Hosts die gleiche IQN verwenden, neu generieren auf `oracdb2` (`stop iscsi`, `clear /var/lib/iscsi/nodes/*`, `new InitiatorName in /etc/iscsi/initiatorname.iscsi`, `restart iscsid`).

GCNV iSCSI-Volumen bereitstellen

Schritt 1: Erstellen Sie einen GCNV Flex Unified iSCSI-Speicherpool pro Datenbankzone

Ein Flex Unified Pool pro Datenbankzone (siehe [Architektur](#)).

Erstellen Sie zwei Pools für dieses HA-Design (wiederholen Sie die Schritte für jede Zone):

Poolname	Zone	Wird verwendet von
oracle-pool-a	us-west1-a	oracdb1 (primär)
oracle-pool-b	us-west1-b	oracdb2 (Standby)

NetApp Volumes → **Speicherpools** → **Erstellen** für jeden Pool:

- **Servicelevel:** Flex (nicht Premium)

- **Typ:** Einheitlich
- **Zone:** stimmt mit der Datenbank-VM-Zone überein (`us-west1-a/ us-west1-b`)
- **Wichtiger Hinweis:** verbunden mit `oracle-vpc`
- **Kapazität:** dimensioniert für die Arbeitslast; verwenden Sie benutzerdefinierten Durchsatz/IOPS, wenn Redo, Backup oder Restore den Standardspielraum überschreiten (bis zu 5120 MiB/s oder 160K IOPS pro Pool, gemäß Produktbeschränkungen)

Warten Sie auf `READY`. Skalieren Sie die Poolgrößen entsprechend Ihrer Datenbank-Footprint (die Größen in [Schritt 3: Erstellen der GCNV iSCSI-Volumes](#) sind Beispiele).



Standardmodus (diese Anleitung): Flex Unified pools verwenden den Standardmodus (`--mode=default`). Erstellen Sie Pools und iSCSI-Volumes mit Cloud Console oder `gcloud netapp`. Volume-Replikation, Snapshots und Klone verwenden Google Cloud APIs ([Schritt 3: GCNV-Standby-Initialisierung](#)).

Schritt 2: Erstellen Sie die Hostgruppen (eine pro DB-Host)

Erstellen Sie eine Hostgruppe pro Datenbank-Host, sodass jede VM nur ihre eigenen Volumes sieht – primärer und Standby-Host dürfen keine GCNV-iSCSI-Volumes gemeinsam nutzen. Der Observer verfügt über keine GCNV-Ressourcen. In der Cloud Console:

1. **NetApp Volumes** → **Hostgruppen** → **Erstellen**
2. **Name:** `oracdb1-hg` · **Region:** `us-west1` · **Typ:** iSCSI-Initiator · **Betriebssystemtyp:** Linux
3. **Hosts:** Fügen Sie den IQN aus `oracdb1` (dem Wert von `/etc/iscsi/initiatorname.iscsi`) ein.
4. **Beschreibung:** „Oracle-Primärhost `oracdb1`“ · **Erstellen**
5. Wiederholen Sie dies für `oracdb2-hg` mit `oracdb2`'s IQN

Schritt 3: Erstellen der GCNV iSCSI-Volumes (pro Datenbank-Host)

Jeder Datenbankhost erhält fünf GCNV iSCSI-Volumes im Pool seiner Zone – eines für `/u01` und vier ASM-Backing-Devices:

GCNV iSCSI-Volume	Größe	Verwenden	Multipath-Alias
<code>ora_<host>_u01</code>	100 GiB	<code>/u01</code> GCNV iSCSI-Volume – Grid/Oracle-Homes, Staging	<code>/dev/mapper/ora_<host>_u01</code>
<code>ora_<host>_data_01</code>	50 GiB	ASM +DATA	<code>/dev/mapper/ora_<host>_data_01</code>
<code>ora_<host>_data_02</code>	50 GiB	ASM +DATA (gestreift)	<code>/dev/mapper/ora_<host>_data_02</code>
<code>ora_<host>_arch_01</code>	100 GiB	ASM +RECO	<code>/dev/mapper/ora_<host>_arch_01</code>
<code>ora_<host>_fra_01</code>	100 GiB	ASM +FRA	<code>/dev/mapper/ora_<host>_fra_01</code>

Volume-Namen: Nur Buchstaben, Zahlen und Unterstriche (keine Bindestriche).



Minimales Layout (nur zur Validierung): Zwei LUNs pro Host (*_data, *_reco) mit arch_01p1→+RECO und arch_01p2→+FRA sind für das Labor akzeptabel; in der Produktion werden fünf Volumes pro [Schritt 3: Erstellen der GCNV iSCSI-Volumes](#) verwendet.

Auf oracdb1: Erstellen Sie alle fünf Volumes in oracle-pool-a, Hostgruppe oracdb1-hg.

Auf oracdb2: Erstellen Sie alle fünf Volumes in oracle-pool-b, Hostgruppe oracdb2-hg.

NetApp Volumes → Volumes → Create — iSCSI, korrekter Pool und Hostgruppe, Linux. Datensatz pro Pool:

- iSCSI-Portal-IPs → <ISCSI_PORTAL_1>, <ISCSI_PORTAL_2> (primäre Pool-Portale auf oracdb1; Standby-Pool-Portale auf oracdb2 — sie können unterschiedlich sein)
- Datenträger-Seriennummer aus der Cloud Console – Verwendung mit der vom Host ermittelten WWID in [Schritt 4: Linux iSCSI und Multipath für GCNV iSCSI-Volumes konfigurieren](#)

Schritt 4: Linux iSCSI und Multipath für GCNV iSCSI-Volumes konfigurieren

Führen Sie den Vorgang zuerst auf oracdb1 unter Verwendung der Poolportale dieses Hosts aus und dann auf oracdb2 unter Verwendung der Standby-Poolportale.

Wenn der ausgehende Datenverkehr des Hosts eingeschränkt ist, erlauben Sie TCP/3260 von jeder Datenbank-VM zu ihren GCNV iSCSI-Portal-IPs (zusätzlich zu inter-VM TCP/1521 von [Schritt 2: VPC-Firewall – TCP/1521 in allen drei Zonen auf die Allowlist setzen](#)).

1. Ziele ermitteln, anmelden und Knotenstart beibehalten:

```
sudo iscsiadm --mode discovery --op update --type sendtargets --portal
<ISCSI_PORTAL_1>
sudo iscsiadm --mode discovery --op update --type sendtargets --portal
<ISCSI_PORTAL_2>
sudo iscsiadm --mode node --op update --name node.startup --value
automatic
sudo iscsiadm --mode node -l all
sudo systemctl enable --now iscsid iscsi multipathd
sudo iscsiadm --mode session          # expect 10 sessions (5 GCNV iSCSI
volumes × 2 portals)
sudo lsblk -o NAME,SIZE,WWN,VENDOR,MODEL
```

Nach dem Neustart vor dem Starten von Oracle erneut prüfen:

+

```
sudo iscsiadm --mode session
sudo multipath -ll
```

1. Konfigurieren device-mapper-multipath:

```

sudo tee /etc/multipath.conf >/dev/null <<'EOF'
defaults {
    find_multipaths      yes
    user_friendly_names yes
}
blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st) [0-9]*"
    devnode "^hd[a-z]"
    devnode "^cciss.*"
}
EOF
+
sudo systemctl enable --now multipathd
sudo multipath -ll

```

1. Füge vom Host ermittelte WWID-Aliase zu /etc/multipath.conf hinzu (nicht raten — multipath.conf erweitert **keine** Shell-Variablen). WWIDs ermitteln:

```

sudo multipath -ll
for dev in /dev/sd*; do
    [ -b "$dev" ] || continue
    printf '%s: ' "$dev"
    sudo /usr/lib/udev/scsi_id --whitelisted --device="$dev" 2>/dev/null
    || true
    echo
done

```

Füge konkrete Aliase für diesen Host zu /etc/multipath.conf hinzu und dann `sudo systemctl restart multipathd`.

An oracdb1, anhängen:

```

multipaths {
    multipath { wwid <host-discovered-wwid-for-u01>      alias
ora_oracdb1_u01      }
    multipath { wwid <host-discovered-wwid-for-data-01>  alias
ora_oracdb1_data_01 }
    multipath { wwid <host-discovered-wwid-for-data-02>  alias
ora_oracdb1_data_02 }
    multipath { wwid <host-discovered-wwid-for-arch-01>  alias
ora_oracdb1_arch_01 }
    multipath { wwid <host-discovered-wwid-for-fra-01>   alias
ora_oracdb1_fra_01  }
}

```

+ Bei oracdb2, verwenden Sie dasselbe Muster mit ora_oracdb2_* Aliassen, dann:

```

sudo systemctl restart multipathd
ls -l /dev/mapper/ora_$(hostname -s)_*

```

Schritt 5: Partitionieren der ASM-Backing-Geräte auf GCNV iSCSI-Volumes

Partitionieren Sie die vier ASM-Backing-Devices (nicht u01) jeweils mit einer GPT-Partition. ASM verwendet die Rohpartition. Führen Sie den Befehl auf jedem Host aus. Alle nachfolgenden Blöcke verwenden `HOST=$(hostname -s)`, um die Geräte des lokalen Hosts automatisch auszuwählen.

```

HOST=$(hostname -s)           # oracdb1 on the primary, oracdb2 on the
standby
for dev in /dev/mapper/ora_${HOST}_data_01 \
           /dev/mapper/ora_${HOST}_data_02 \
           /dev/mapper/ora_${HOST}_arch_01 \
           /dev/mapper/ora_${HOST}_fra_01; do
    sudo parted -s "$dev" mklabel gpt
    sudo parted -s "$dev" mkpart primary 0% 100%
done
sudo partprobe
sudo systemctl reload multipathd
ls /dev/mapper/ora_${HOST}*_p1      # expect 4 partitions

HOST=$(hostname -s)
sudo tee /etc/udev/rules.d/99-oracle-asm.rules >/dev/null <<'EOF'
KERNEL=="dm-*", ENV{DM_UUID}=="part?-mpath-*",
ENV{DM_NAME}=="ora_oracdb*_p?", \
    OWNER="grid", GROUP="asmadmin", MODE="0660"
EOF

sudo udevadm control --reload-rules
for part in /dev/mapper/ora_${HOST}*_p1; do
    dm=$(readlink -f "$part" | xargs basename)
    sudo udevadm trigger --action=change --name-match="/dev/${dm}"
done
sudo udevadm settle
ls -lL /dev/mapper/ora_${HOST}*_p1  # grid:asmadmin 0660

```

Schritt 6: Formatieren und Einbinden /u01 auf dem lokalen GCNV iSCSI-Volume

Das ora_<host>_u01 GCNV iSCSI-Volume enthält Grid-Home, Oracle-Home und Staging. Formatieren Sie XFS auf dem Multipath-Gerät (nicht partitioniert für ASM). Verwenden Sie die UUID in /etc/fstab (keine gemeinsam genutzte Dateisystembezeichnung):

```

HOST=$(hostname -s)
U01_DEV=/dev/mapper/ora_${HOST}_u01
ls -l "$U01_DEV"

sudo mkfs.xfs -f "$U01_DEV"
U01_UUID=$(sudo blkid -s UUID -o value "$U01_DEV")

sudo mkdir -p /u01
echo "UUID=${U01_UUID} /u01 xfs defaults,_netdev,nofail,x-
systemd.requires=iscsi.service,x-systemd.requires=multipathd.service,x-
systemd.after=iscsi.service,x-systemd.after=multipathd.service 0 0" | sudo
tee -a /etc/fstab
sudo mount -a

sudo mkdir -p /u01/app/oraInventory /u01/app/26ai/grid /u01/app/grid \
/u01/app/oracle/product/26ai/db_1 /u01/stage
sudo chown -R grid:oinstall /u01/app/oraInventory /u01/app/26ai
/u01/app/grid
sudo chown -R oracle:oinstall /u01/app/oracle /u01/stage
sudo chmod -R 775 /u01/app /u01/stage

```

Starten Sie das System einmal neu und bestätigen Sie `/u01` die Mounts, bevor [Installieren Sie die Oracle-Software](#).

Installieren Sie die Oracle-Software

Führen Sie zunächst die Grid- und dann die Datenbank-Home-Installation auf jedem DB-Host durch, bevor [Erstellen Sie die primäre Datenbank](#).

Schritt 1: Installieren Sie Oracle Grid Infrastructure (Oracle Restart) auf jedem DB-Host

Führen Sie diesen gesamten Abschnitt auf `oracdb1` aus und wiederholen Sie ihn dann auf `oracdb2`. Beide Hosts erhalten ihr eigenes Grid-Home, ihre eigene ASM-Instanz und ihre eigenen Festplattengruppen – Data Guard repliziert über Oracle Net, nicht über den Speicher.

Stellen Sie die Oracle-Binärdateien auf /u01 bereit

```

sudo chown oracle:oinstall /u01/stage && sudo chmod 775 /u01/stage
# Upload GoldImages, RU, OPatch to /u01/stage.

```

Entpacken Sie das Grid-Startverzeichnis an Ort und Stelle

Das 26ai Grid GoldImage wird installiert, indem es im Zielverzeichnis des Grids entpackt wird:

```

sudo -u grid bash -c '
cd /u01/app/26ai/grid
unzip -q /u01/stage/LINUX.X64_<RELEASE>_grid_home.zip
'
sudo chown -R grid:oinstall /u01/app/26ai/grid

```

Grid-RU-Level. Diese Anleitung setzt voraus, dass das Grid GoldImage bereits Ihre validierte RU enthält. Wenn das Grid GoldImage älter als die Ziel-RU ist, patchen Sie das Grid-Verzeichnis während der Einrichtung gemäß dem von Oracle dokumentierten `gridSetup.sh -applyRU` Ablauf, oder verwenden Sie ein Grid GoldImage mit integrierter RU. Halten Sie Grid- und Datenbank-Verzeichnisse auf demselben vorgesehenen Patch-Level.

Einzelschuss `gridSetup` – vollständige `HA_CONFIG`-Antwortdatei

Build `/tmp/grid.rsp` auf jedem Host (`responseFileVersion` ist obligatorisch; ersetzen `HOST` und verwenden Sie starke Passwörter:

```

HOST=$(hostname -s)

sudo -u grid bash -c "cat > /tmp/grid.rsp <<RSP
oracle.install.responseFileVersion=/oracle/install/rspfmt_crsinstall_response_schema_v23.0.0
INVENTORY_LOCATION=/u01/app/oraInventory
installOption=HA_CONFIG
ORACLE_BASE=/u01/app/grid
clusterUsage=GENERAL_PURPOSE
OSDBA=asmdba
OSOPER=asmoper
OSASM=asmadmin
storageOption=FLEX_ASM_STORAGE
sysasmPassword=WelcomeOracle1!
asmsnmpPassword=WelcomeOracle1!
diskGroupName=DATA
redundancy=EXTERNAL
auSize=4
diskString=/dev/mapper/ora_${HOST}_*p*
diskList=/dev/mapper/ora_${HOST}_data_01p1,/dev/mapper/ora_${HOST}_data_02p1
managementOption=NONE
RSP"
sudo -u grid chmod 600 /tmp/grid.rsp

```

Führen Sie `gridSetup` aus, um die Binärdateien zu kopieren und die Konfiguration bereitzustellen:

```
sudo -u grid bash -c '  
export ORACLE_HOME=/u01/app/26ai/grid  
export ORACLE_BASE=/u01/app/grid  
cd /u01/app/26ai/grid  
./gridSetup.sh -silent -responseFile /tmp/grid.rsp -ignorePrereqFailure  
'
```

Erwarten Sie Successfully Setup Software with warning(s) . und Exit-Code 6 (Warnungen) oder 0.

Führen Sie orainstRoot.sh und root.sh als Root aus

```
sudo /u01/app/oraInventory/orainstRoot.sh  
sudo /u01/app/26ai/grid/root.sh
```

root.sh erstellt die crsctl / srvctl / asmcmd Wrapper und startet OHAS.

gridSetup.sh -executeConfigTools — **ASM aufrufen und erstellen** +DATA

Führen Sie die Konfigurationsassistenten (NETCA, ASMCA, CVU) anhand der Antwortdatei aus — dadurch werden die ASM-Instanz und die +DATA Festplattengruppe erstellt:

```
sudo -u grid bash -c '  
export ORACLE_HOME=/u01/app/26ai/grid  
export ORACLE_BASE=/u01/app/grid  
cd /u01/app/26ai/grid  
./gridSetup.sh -silent -executeConfigTools -responseFile /tmp/grid.rsp  
'
```

Erwarten Sie Successfully Configured Software. nach NETCA / ASMCA / CVU.

Erstellen +RECO und +FRA Festplattengruppen

Die einmalige Installation erstellt nur +DATA. Die anderen beiden erstellen Sie über asmca:

```

HOST=$(hostname -s)

sudo -u grid bash -c "
export ORACLE_HOME=/u01/app/26ai/grid
export ORACLE_SID=+ASM

\${ORACLE_HOME}/bin/asmca -silent -createDiskGroup \
  -diskGroupName RECO \
  -disk /dev/mapper/ora_${HOST}_arch_01p1 \
  -redundancy EXTERNAL -au_size 4

\${ORACLE_HOME}/bin/asmca -silent -createDiskGroup \
  -diskGroupName FRA \
  -disk /dev/mapper/ora_${HOST}_fra_01p1 \
  -redundancy EXTERNAL -au_size 4
"

```

Überprüfen Sie ASM und Oracle Restart

```

sudo -u grid ORACLE_HOME=/u01/app/26ai/grid ORACLE_SID=+ASM \
  /u01/app/26ai/grid/bin/sqlplus -s / as sysasm <<'SQL'
SELECT name, total_mb, free_mb, state FROM v$asm_diskgroup ORDER BY name;
SQL

sudo /u01/app/26ai/grid/bin/crsctl stat res -t
# Expected ONLINE: ora.DATA.dg, ora.RECO.dg, ora.FRA.dg,
ora.LISTENER.lsnr, ora.asm, ora.cssd, ora.evmd.

```

Wiederholen Sie diesen Abschnitt auf `oracdb2`. Das `HOST=$(hostname -s)`-Muster in und wählt die GCNV iSCSI-Geräte dieses Hosts automatisch aus. Verwenden Sie dieselben ASM-Festplattengruppennamen — Data Guard repliziert über Oracle Net, nicht über den Speicher.

Schritt 2: Installieren Sie Oracle Database 26ai auf jedem DB-Host

Führen Sie diesen Abschnitt auf `oracdb1` aus und dann auf `oracdb2`. Die Standby-Datenbank wird in [Erstellen Sie die Standby-Datenbank](#) erstellt.

Entpacken Sie das DB-Home und den RU-Patch

```

sudo su - oracle
cd /u01/app/oracle/product/26ai/db_1
unzip -q /u01/stage/LINUX.X64_<RELEASE>_db_home.zip
rm -rf OPatch
unzip -q /u01/stage/p6880880_<base>_Linux-x86-64.zip #
latest OPatch
unzip -q /u01/stage/p<RU_PATCH>_<base>_Linux-x86-64.zip -d /u01/stage #
latest 26ai RU

```

Informationen zum RU-Verzeichnisaufbau und `-applyRU`-Pfad finden Sie in der Oracle-Dokumentation.

Stille, rein softwarebasierte Installation mit angewendeter RU

```

sudo -u oracle tee /u01/stage/dbinstall.rsp >/dev/null <<'EOF'
oracle.install.option=INSTALL_DB_SWONLY
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
ORACLE_BASE=/u01/app/oracle
oracle.install.db.InstallEdition=EE
oracle.install.db.OSDBA_GROUP=dba
oracle.install.db.OSOPER_GROUP=oper
oracle.install.db.OSBACKUPDBA_GROUP=backupdba
oracle.install.db.OSDGDBA_GROUP=dgdba
oracle.install.db.OSKMDBA_GROUP=kmdba
oracle.install.db.OSRACDBA_GROUP=racdba
oracle.install.db.rootconfig.executeRootScript=false
EOF

sudo -u oracle bash -c '
export CV_ASSUME_DISTID=OEL10 # OEL9 / OEL8.10 if cluvfy requires it
cd /u01/app/oracle/product/26ai/db_1
./runInstaller -applyRU /u01/stage/<RU_PATCH> \
  -applyOneOffs /u01/stage/39292021 \
  -silent -ignorePrereqFailure -responseFile /u01/stage/dbinstall.rsp
'

```

Bei OL 8/9 `-applyOneOffs` aus der `runInstaller` Zeile weglassen.

Als root führen Sie das Nachinstallationskript aus:

```

sudo /u01/app/oracle/product/26ai/db_1/root.sh

```

Oracle-Umgebung einrichten

Auf jedem DB-Host (orcl auf oracdb1, orcls auf oracdb2):

```
sudo -u oracle tee -a /home/oracle/.bash_profile >/dev/null <<'EOF'
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export ORACLE_SID=orcl # use 'orcls' on oracdb2
export GRID_HOME=/u01/app/26ai/grid
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH
export TNS_ADMIN=$ORACLE_HOME/network/admin
EOF
```

(Verwenden Sie ORACLE_SID=orcls auf dem Standby-Host. Die Standby-Datenbank wird in [Erstellen Sie die Standby-Datenbank](#) erstellt.)

Erstellen Sie die primäre Datenbank (oracdb1 nur)

Im stillen Modus dbca gegen die ASM-Festplattengruppen ausführen:

```
sudo -u oracle bash -c '
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export PATH=$ORACLE_HOME/bin:$PATH

dbca -silent -createDatabase \
  -templateName General_Purpose.dbc \
  -gdbname orcl -sid orcl \
  -characterSet AL32UTF8 -nationalCharacterSet AL16UTF16 \
  -sysPassword "ChangeMe!1" -systemPassword "ChangeMe!1" \
  -emConfiguration NONE \
  -datafileDestination +DATA -storageType ASM \
  -recoveryAreaDestination +FRA -recoveryAreaSize 25000 \
  -enableArchive true -archiveLogMode AUTO \
  -memoryMgmtType AUTO_SGA -totalMemory 4096 \
  -databaseType MULTIPURPOSE \
  -createAsContainerDatabase true -numberOfPDBs 1 \
  -pdbName orclpdb -pdbAdminPassword "ChangeMe!1" \
  -ignorePreReqs
'
```

Archivprotokolle auf +RECO verweisen (der Standby verwendet übereinstimmende Einstellungen in [Schritt 2: Standby: Minimale init.ora pfile, Passwortdatei, NOMOUNT](#)):

```

sudo -u oracle bash -c '
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export ORACLE_SID=orcl
$ORACLE_HOME/bin/sqlplus -s / as sysdba <<SQL
ALTER SYSTEM SET log_archive_dest_1='\'LOCATION=+RECO
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=orcl\'\' SCOPE=BOTH;
ALTER PLUGGABLE DATABASE ALL OPEN;
ALTER PLUGGABLE DATABASE ALL SAVE STATE;
EXIT
SQL
'
```

Überprüfen Sie, ob die Datenbank unter Oracle Restart verfügbar ist:

```

sudo /u01/app/26ai/grid/bin/srvctl status database -d orcl
# Expected: Database is running

sudo -u oracle sqlplus -s / as sysdba <<<"SELECT name, open_mode, log_mode
FROM v\${database};"
# Expected: ORCL, READ WRITE, ARCHIVELOG
```

Erstellen Sie einen rollenbasierten Anwendungsdienst (Oracle Restart). Anwendungen sollten sich über orclapp und nicht über den Datenbanknamen verbinden, damit ein Failover transparent erfolgt.

```

sudo -u oracle bash -c '
export GRID_HOME=/u01/app/26ai/grid
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH

srvctl add service \
  -db orcl \
  -service orclapp \
  -pdb orclpdb \
  -role PRIMARY \
  -policy AUTOMATIC

srvctl start service -db orcl -service orclapp
srvctl status service -db orcl -service orclapp
'
```

Nach der Broker-Aktivierung, orclapp läuft nur auf PRIMARY. Steuerdateien über ASM-Festplattengruppen multiplexen; Arbeitsspeicher auf die Arbeitslast dimensionieren (in diesem Leitfaden werden 4 GB / 3 GB SGA als Beispiele verwendet).

Erstellen Sie die Standby-Datenbank

Build `orcls` on `oracdb2` (dedizierte Volumes auf `oracle-pool-b`). Schließen Sie die anfängliche Einrichtung von Primary und Standby ab, dann [Schritt 3: GCNV-Standby-Initialisierung](#), die Standby-Abschlussaufgaben und [Konfigurieren Sie Data Guard Broker, FSFO und Observer](#). Ziel: Primary READ WRITE; Standby PHYSICAL STANDBY, MOUNTED, MRP applying.

Schritt 1: Primär: SYS-Passwort, Passwortdatei und DG-Parameter

An `oracdb1` als `oracle`:

```
sudo su - oracle
. ~/.bash_profile          # ORACLE_SID=orcl, ORACLE_HOME set
```

An `oracdb2`:

```
GRID_HOME=/u01/app/26ai/grid
sudo -u grid tee "$GRID_HOME/network/admin/listener.ora" >/dev/null <<'
EOF'
LISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = oracdb2.example.internal) (PORT =
1521)))

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC = (GLOBAL_DBNAME = orcls)          (ORACLE_HOME =
/u01/app/oracle/product/26ai/db_1) (SID_NAME = orcls))
    (SID_DESC = (GLOBAL_DBNAME = orcls_DGMGRL) (ORACLE_HOME =
/u01/app/oracle/product/26ai/db_1) (SID_NAME = orcls)))
EOF
```

Neustart über Oracle Restart auf jedem Host:

```
sudo -u grid bash -c '
export GRID_HOME=/u01/app/26ai/grid
export ORACLE_HOME=$GRID_HOME
$GRID_HOME/bin/srvctl stop listener
$GRID_HOME/bin/srvctl start listener
$GRID_HOME/bin/lsnrctl status
'
```

`lsnrctl status`muss `<SID>`auflisten und `<SID>_DGMGRL.`

Schritt 2: Standby: Minimale init.ora pfile, Passwortdatei, NOMOUNT

Kopieren Sie die primäre Kennwortdatei auf den Standby (IAP `gcloud compute scp`):

```
PRIMARY_ZONE=us-west1-a      # zone of oracdb1
STANDBY_ZONE=us-west1-b     # zone of oracdb2

gcloud compute scp \
  oracdb1:/u01/app/oracle/product/26ai/db_1/dbs/orapworcl ./orapworcl \
  --zone=${PRIMARY_ZONE} --tunnel-through-iap

gcloud compute scp \
  ./orapworcl oracdb2:/u01/app/oracle/product/26ai/db_1/dbs/orapworcls \
  --zone=${STANDBY_ZONE} --tunnel-through-iap
```

Auf `oracdb2` dem System die Besitzrechte festlegen und die Standby-PFILE erstellen. Auf dem **primären** `*.compatible` zuerst kopieren:

```
# On oracdb1
sudo -u oracle sqlplus -s / as sysdba \
  <<<"SELECT value FROM v\parameter WHERE name='compatible';"
```

Am `oracdb2` ersetzen Sie diesen Wert durch `<COPY_FROM_PRIMARY>` im folgenden Block:

```

sudo -u oracle mkdir -p /u01/app/oracle/admin/orcls/adump
sudo chown oracle:oinstall
/u01/app/oracle/product/26ai/db_1/dbs/orapworcls
sudo chmod 0600 /u01/app/oracle/product/26ai/db_1/dbs/orapworcls

sudo -u oracle tee /u01/app/oracle/product/26ai/db_1/dbs/initorcls.ora
>/dev/null <<'EOF'
*.db_name='orcl'
*.db_unique_name='orcls'
*.audit_file_dest='/u01/app/oracle/admin/orcls/adump'
*.diagnostic_dest='/u01/app/oracle'
*.compatible='<COPY_FROM_PRIMARY>'
*.sga_target=3072m
*.pga_aggregate_target=1024m
*.processes=320
*.remote_login_passwordfile='EXCLUSIVE'
*.standby_file_management='AUTO'
*.fal_server='orcl'
*.log_archive_config='DG_CONFIG=(orcl,orcls)'
*.log_archive_dest_1='LOCATION=+RECO VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=orcls'
*.log_archive_dest_2='SERVICE=orcl AFFIRM SYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=orcl'
*.log_archive_dest_state_2='DEFER'
*.log_archive_format='%t_%s_%r.arc'
*.dg_broker_start=TRUE
*.undo_tablespace='UNDOTBS1'
*.open_cursors=300
*.db_create_file_dest='+DATA'
*.db_create_online_log_dest_1='+DATA'
*.db_recovery_file_dest='+FRA'
*.db_recovery_file_dest_size=25000m
EOF

echo "orcls:/u01/app/oracle/product/26ai/db_1:N" | sudo tee -a /etc/oratab

sudo -u oracle bash -c '
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export ORACLE_SID=orcls
sqlplus / as sysdba <<SQL
STARTUP NOMOUNT PFILE=/u01/app/oracle/product/26ai/db_1/dbs/initorcls.ora;
EXIT
SQL
'
```

NOMOUNT ohne Datendateien bis [Schritt 3: GCNV-Standby-Initialisierung](#).

Schritt 3: GCNV-Standby-Initialisierung

Füllen Sie Standby-ASM-Volumes in `oracle-pool-b` mithilfe der **Google Cloud NetApp Volumes-Replikation** (SnapMirror-basiert, **Default-mode**), **Volume-Snapshots** oder **Klon**, und führen Sie dann [Oracle finalisiert](#) aus. iSCSI und ASM sind identisch mit [GCNV iSCSI-Volumes bereitstellen](#).

API	Verwenden
<code>gcloud netapp storage-pools</code>	Erstellen von Standardmodus-Flex Unified-Pools (<code>--mode=default</code>)
<code>gcloud netapp volumes</code>	iSCSI-Volumes, Hostgruppen, Snapshots
<code>gcloud netapp volumes replications</code>	Standortübergreifende Volume-Replikation

Schritt	Aktion
1	Primäres Archivprotokoll + erzwungene Protokollierung (Schritt 1: Primär: SYS-Passwort, Passwortdatei und DG-Parameter)
2	Ruhezustand: <code>BEGIN BACKUP</code> , SCN aufzeichnen, Standby-Kontrolldatei
3	Volumenreplikation (Volume-Replikationen erstellen und Cutover – Ruhezustand, Stopp der Replikation, Anschließen von Standby-LUNs) oder Snapshot (Alternative — Snapshot-Seed)
4	On <code>oracdb2</code> : iSCSI, Multipath, ASM-Mount (Schritt 4: Linux iSCSI und Multipath für GCNV iSCSI-Volumes konfigurieren , Schritt 5: Partitionieren der ASM-Backing-Geräte auf GCNV iSCSI-Volumes und)
5	Oracle finalize — Wiederherstellung zu SCN, MOUNTED (Oracle finalisiert)
6	SRL-Neuaufbau (Schritt 4: Standby-Wiederherstellungsprotokolldateien), MRP, Broker (Konfigurieren Sie Data Guard Broker, FSFO und Observer)

Die aktive RMAN-Duplikation bleibt für kleine Labore gültig. **GCNV-Replikation wird für die Bereitstellung von Produktions-Standby-Systemen bevorzugt.**

Voraussetzungen

- `gcloud netapp` mit Unterstützung für Volumenreplikation.
- Zwei **Standardmodus**-Pools an verschiedenen Standorten (`oracle-pool-a`, `oracle-pool-b`).
- Quellvolumes im primären Pool sind an `oracdb1-hg`; Zielvolumes werden durch Replikation erstellt.
- Führen Sie die Replikation von der Cloud Shell oder einer Workstation aus – nicht von den DB-VMs.

```
export PROJECT=<your-gcp-project>
export LOC_A=us-west1-a
export LOC_B=us-west1-b
export DEST_POOL="projects/${PROJECT}/locations/${LOC_B
}/storagePools/oracle-pool-b"
```

Erstellen Sie bei Bedarf einen Standby-Pool:

```
gcloud netapp storage-pools create oracle-pool-b \
  --project="${PROJECT}" --location="${LOC_B}" \
  --service-level=flex --type=unified --mode=default \
  --capacity=1024 --network=name=<your-vpc>
```

Volume-Replikationen erstellen

```
gcloud netapp volumes replications create repl-oracdb2-data \
  --project="${PROJECT}" --location="${LOC_A}" --volume=oracdb1_data \
  --replication-schedule=EVERY_10_MINUTES \
  --destination-volume-parameters="storage_pool=${DEST_POOL
},volume_id=oracdb2_data,share_name=oracdb2_data"

gcloud netapp volumes replications create repl-oracdb2-reco \
  --project="${PROJECT}" --location="${LOC_A}" --volume=oracdb1_reco \
  --replication-schedule=EVERY_10_MINUTES \
  --destination-volume-parameters="storage_pool=${DEST_POOL
},volume_id=oracdb2_reco,share_name=oracdb2_reco"
```

Warten Sie, bis `mirrorState` **MIRRORED** / die anfängliche Synchronisierung abgeschlossen ist.

Cutover – Ruhezustand, Stopp der Replikation, Anschließen von Standby-LUNs

Primär:

```
ALTER DATABASE BEGIN BACKUP;
SELECT CURRENT_SCN FROM V$DATABASE;
ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/tmp/orcls_stby.ctl';
```

Lassen Sie den letzten Replikationszyklus zu, dann:

```
gcloud netapp volumes replications stop repl-oracdb2-data \  
  --project="${PROJECT}" --location="${LOC_A}" --volume=oracdb1_data  
--force
```

```
gcloud netapp volumes replications stop repl-oracdb2-reco \  
  --project="${PROJECT}" --location="${LOC_A}" --volume=oracdb1_reco  
--force
```

Zielvolumes an oracdb2-hg anhängen (replizierte LUN behält möglicherweise den Quellnamen bei – name=oracdb1_data_lun bei Aktualisierung verwenden):

```
HG=$(gcloud netapp host-groups describe oracdb2-hg --project="${PROJECT}" \  
 \  
  --location=us-west1 --format='value(name) ')  
  
gcloud netapp volumes update oracdb2_data --project="${PROJECT}" \  
--location="${LOC_B}" \  
--block-devices="name=oracdb1_data_lun,host-groups=${HG},os-type=LINUX"
```

Kopiere die Kontrolldatei nach oracdb2, dann auf dem primären System:

```
ALTER DATABASE END BACKUP;
```

Alternative — Snapshot-Seed

Einmaliger Seed: Snapshot auf dem Quellvolume → Volume aus dem Snapshot im Standby-Pool erstellen (Cloud Console oder API). Fahren Sie nach [Oracle finalisiert](#) fort, nachdem Sie an oracdb2-hg angehängt haben.

Standby iSCSI und ASM (vor RMAN)

Auf oracdb2 melden Sie sich an den **Standby-Pool**-iSCSI-Portalen an. Wenn die ASM-Disk-Header mit der primären Namensgebung übereinstimmen, verwenden Sie **Multipath-Aliase im primären Stil** (Lab: ora_oracdb1_data_01, ora_oracdb1_arch_01), setzen Sie `asm_diskstring='/dev/mapper/ora_oracdb1_*p*', chown grid:asmadmin` auf Partitionen und dann:

```
ALTER DISKGROUP DATA MOUNT FORCE;  
ALTER DISKGROUP RECO MOUNT FORCE;  
ALTER DISKGROUP FRA MOUNT FORCE;
```

Oracle finalisiert

```

STARTUP NOMOUNT;
RESTORE STANDBY CONTROLFILE FROM '/tmp/orcls_stby.ctl';
ALTER DATABASE MOUNT;
RECOVER DATABASE UNTIL SCN <quiesce_scn>;
ALTER DATABASE CONVERT TO PHYSICAL STANDBY;
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;

```

Nach der Konvertierung **Standby-Redo-Logs neu erstellen** (die replizierte Kontrolldatei enthält noch +DATA/ORCL/... SRL-Pfade — verursacht ORA-19527 / ORA-16086 auf dem primären System). Siehe [Schritt 4: Standby-Wiederherstellungsprotokolldateien](#).

Schritt 4: Standby-Wiederherstellungsprotokolldateien

Erforderlich auf **beiden** Hosts für FSFO. Größe ≥ größte primäre Online-Redo; Anzahl = (Online-Gruppen pro Thread) + 1.

Nach GCNV-Seed: Alle Standby-Logdateigruppen auf dem Standby-System löschen und auf +DATA`nur (`db_create_file_dest='+DATA' neu erstellen). Replizierte Pfade unter +DATA/ORCL/... verursachen ORA-19527/ ORA-16086 bis zur Wiederherstellung.

Primär (orcl):

```

ALTER SYSTEM SET db_create_file_dest='+DATA' SCOPE=BOTH;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 ('+DATA') SIZE 1024M;
-- repeat (online log groups + 1) times

```

Standby (orcls) nach GCNV-Seed:

```

ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
ALTER SYSTEM SET standby_file_management=MANUAL SCOPE=BOTH;
-- DROP STANDBY LOGFILE GROUP for each group# in v$standby_log;
ALTER SYSTEM SET db_create_file_dest='+DATA' SCOPE=BOTH;
ALTER SYSTEM SET standby_file_management=AUTO SCOPE=BOTH;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 ('+DATA') SIZE 1024M;
-- repeat (online groups + 1) times; one member per group
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE
DISCONNECT FROM SESSION;

```

Schritt 5: Flashback auf dem Standby-System aktivieren und verwaltete Wiederherstellung starten

Flashback muss **vor** dem Start der verwalteten Wiederherstellung aktiviert werden (Flashback kann nicht aktiviert werden, während MRP aktiv ist).

```
# On oracdb2
sudo -u oracle bash -c '
. ~/.bash_profile
export ORACLE_SID=orcl
sqlplus / as sysdba <<SQL
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER SYSTEM SET db_flashback_retention_target=1440 SCOPE=BOTH;
ALTER DATABASE FLASHBACK ON;
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE
DISCONNECT FROM SESSION;
EXIT
SQL'
```

„USING CURRENT LOGFILE“ ermöglicht die Echtzeit-Anwendung (Wiederherstellungsprotokoll wird angewendet, sobald es in den SRLs landet).

Schritt 6: Wiederherstellungsprotokoll-Versand auf dem primären Server aktivieren



„LOG_ARCHIVE_DEST_2“ wurde absichtlich auf „DEFER“ in [Schritt 2: Standby: Minimale init.ora pfile, Passwortdatei, NOMOUNT](#) gesetzt, um fortlaufende „ORA-12154“ Fehler während der Standby-Erstellung zu unterdrücken. Aktivieren Sie es jetzt, da der Standby bereit ist.

Wechseln Sie LOG_ARCHIVE_DEST_STATE_2 zu „ENABLE“ und erzwingen Sie einen Protokollwechsel, damit die erste Runde des Wiederherstellungsprotokolls sofort übertragen wird:

```
sudo -u oracle bash -c '
. ~/.bash_profile
sqlplus / as sysdba <<SQL
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE SCOPE=BOTH;
ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM ARCHIVE LOG CURRENT;
SELECT dest_id, status, error FROM v$archive_dest_status WHERE dest_id IN
(1,2);
EXIT
SQL
'
```

Expected: dest_id=2, STATUS=VALID, ERROR null.

Falls dest_2 ORA-12154 angezeigt wird, den Primary neu starten. Nach [Schritt 1: Aktivieren Sie den Broker auf beiden Datenbanken](#) den Transport über DGMGRL verwalten.

Schritt 7: Überprüfen Sie den Ziel-Data-Guard-Status

Auf der **primären** (oracdb1):

```
sudo -u oracle sqlplus -s / as sysdba \
  <<<"SELECT database_role || ' | ' || open_mode FROM v\$database;"
# Expected: PRIMARY | READ WRITE
```

Im **Standby-Modus** (oracdb2 — Cloud Console **SSH** oder IAP von Ihrer Workstation aus:

```
gcloud compute ssh oracdb2 --tunnel-through-iap --zone=us-west1-b

sudo -u oracle bash <<'BASH'
. ~/.bash_profile
export ORACLE_SID=orcls

sqlplus -s / as sysdba <<'SQL'
SELECT database_role || ' | ' || open_mode
FROM v$database;

SELECT process, status, sequence#
FROM v$managed_standby
WHERE process IN ('MRP0','RFS');

EXIT
SQL
BASH
```

Erwartet im Standby-Modus: PHYSICAL STANDBY | MOUNTED; MRP0 mit APPLYING_LOG.

Wenn der Standby-Prozess MOUNTED meldet, die Anwendung aber nicht ausgeführt wird, starten Sie MRP auf oracdb2 neu:

```
sudo -u oracle bash -c '
. ~/.bash_profile
export ORACLE_SID=orcls
sqlplus / as sysdba <<SQL
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE
DISCONNECT FROM SESSION;
EXIT
SQL'
```

Schritt 8: Registrieren Sie die Standby-Datenbank bei Oracle Restart

Registrieren Sie den Standby mit Restart nach GCNV-Seeding, damit Neustarts ASM, MOUNT und apply wiederherstellen.

Am `oracdb2` erfassen Sie den Speicherort der spfile und registrieren Sie diesen bei Oracle Restart (ersetzen Sie `<STANDBY_SPFILE_PATH>` aus der Abfrage, oft unter `+DATA`):

```
sudo -u oracle bash -c '
export ORACLE_SID=orcls
sqlplus -s / as sysdba <<< "SHOW PARAMETER spfile;"
'

sudo -u oracle bash -c '
export GRID_HOME=/u01/app/26ai/grid
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH

srvctl add database \
  -db orcls \
  -dbname orcl \
  -oraclehome /u01/app/oracle/product/26ai/db_1 \
  -spfile <STANDBY_SPFILE_PATH> \
  -pwfile /u01/app/oracle/product/26ai/db_1/dbs/orapworcls \
  -role PHYSICAL_STANDBY \
  -startoption MOUNT \
  -stopoption IMMEDIATE \
  -diskgroup DATA,RECO,FRA

srvctl config database -db orcls
srvctl status database -db orcls
'
```

Überprüfen Sie auf `oracdb1`, ob die primäre Oracle Restart Datenbankressource die ASM Festplattengruppen-Abhängigkeiten auflistet. Fügen Sie `RECO` hinzu, wenn nur `DATA` durch DBCA registriert wurde und `FRA`:

```
sudo -u oracle bash -c '
export GRID_HOME=/u01/app/26ai/grid
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH
srvctl config database -db orcl
srvctl modify database -db orcl -diskgroup DATA,RECO,FRA
srvctl config database -db orcl
'
```

Fügen Sie denselben Anwendungsdienst auf der Standby-Datenbankressource (`orcls` hinzu (auf `oracdb2`). Verwenden Sie `role PRIMARY` auf beiden Seiten, damit `orclapp` nach dem Switchover verfügbar ist:

```

sudo -u oracle bash -c '
export GRID_HOME=/u01/app/26ai/grid
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH

srvctl add service \
  -db orcls \
  -service orclapp \
  -pdb orclpdb \
  -role PRIMARY \
  -policy AUTOMATIC

srvctl config service -db orcls -service orclapp
'
```

Bestätigen Sie auf oracdb2 die Standby-Datenbankressource:

```

sudo -u oracle bash -c '
export GRID_HOME=/u01/app/26ai/grid
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH
srvctl status database -db orcls
'
```

Konfigurieren Sie Data Guard Broker, FSFO und Observer

Nach ENABLE CONFIGURATION`der Einrichtung verwalten Sie Transport und Rollen über **DGMGRL** (nicht ad-hoc `LOG_ARCHIVE_DEST_* SQL).

Schritt 1: Aktivieren Sie den Broker auf beiden Datenbanken

Auf den **primären** und **Standby** Datenbank-Hosts:

```

sudo -u oracle bash -c '
. ~/.bash_profile
sqlplus / as sysdba <<SQL
ALTER SYSTEM SET dg_broker_start=TRUE SCOPE=BOTH;
EXIT
SQL'
```

Stellen Sie auf dem **primären Datenbank-Host** eine Verbindung mit Betriebssystemauthentifizierung her (die Observer-Wallet von [Schritt 5: Starten Sie den Observer als systemd-Einheit](#) ist auf DB-Hosts nicht erforderlich):

```
sudo -u oracle bash -c '  
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1  
export ORACLE_SID=orcl  
export PATH=$ORACLE_HOME/bin:$PATH  
dgmgrl /  
'
```



Nur auf dem Observer-Host verwenden Sie `dgmgrl /@orcl` nachdem die Auto-Login-Wallet existiert. Geben Sie keine Passwörter in die `dgmgrl` Befehlszeile ein.

```
DGMGRL> CREATE CONFIGURATION 'orcl_dg' AS  
        PRIMARY DATABASE IS 'orcl' CONNECT IDENTIFIER IS orcl;  
DGMGRL> ADD DATABASE 'orcls' AS CONNECT IDENTIFIER IS orcls;  
DGMGRL> ENABLE CONFIGURATION;  
DGMGRL> SHOW CONFIGURATION;  
-- Expect: Configuration Status: SUCCESS, both members SUCCESS.
```

Führen Sie `VALIDATE DATABASE` sofort aus – es deckt verwaiste Datendateien, fehlende SRLs, nicht laufende Redo-Appl-Vorgänge und andere häufige Fallstricke auf, bevor Sie einen Switchover durchführen. Jede `WARNING` oder nicht-`NULL ERROR` muss vor [Schritt 3: FSFO-Eigenschaften konfigurieren und aktivieren](#) behoben werden.

```
DGMGRL> VALIDATE DATABASE 'orcls';  
DGMGRL> SHOW CONFIGURATION VERBOSE;
```

Schritt 2: Flashback bestätigen (erforderlich für die automatische Wiederherstellung durch FSFO)

Bestätigen Sie `flashback_on` auf **beiden** Hosts, bevor Sie FSFO aktivieren:

```
sudo -u oracle bash -c '  
. ~/.bash_profile  
sqlplus -s / as sysdba <<<"SELECT flashback_on FROM v\${database};"  
'  
# Expected on both hosts: YES
```

Nur auf dem **primären** System, falls die Aufbewahrung nicht bereits festgelegt ist:

```

sudo -u oracle bash -c '
. ~/.bash_profile
export ORACLE_SID=orcl
sqlplus / as sysdba <<SQL
ALTER SYSTEM SET db_flashback_retention_target=1440 SCOPE=BOTH;
EXIT
SQL'

```

Schritt 3: FSFO-Eigenschaften konfigurieren und aktivieren

```

-- Transport mode and protection mode
DGMGRL> EDIT DATABASE 'orcl' SET PROPERTY LogXptMode='SYNC';
DGMGRL> EDIT DATABASE 'orcls' SET PROPERTY LogXptMode='SYNC';
DGMGRL> EDIT CONFIGURATION SET PROTECTION MODE AS MaxAvailability;

-- FSFO targets (each side names the other)
DGMGRL> EDIT DATABASE 'orcl' SET PROPERTY FastStartFailoverTarget =
'orcls';
DGMGRL> EDIT DATABASE 'orcls' SET PROPERTY FastStartFailoverTarget =
'orcl';

-- 30 s = default; lower for faster RTO but more sensitive to network
blips
DGMGRL> EDIT CONFIGURATION SET PROPERTY FastStartFailoverThreshold = 30;
DGMGRL> EDIT CONFIGURATION SET PROPERTY FastStartFailoverAutoReinstate =
TRUE;

DGMGRL> ENABLE FAST_START FAILOVER;
DGMGRL> SHOW FAST_START FAILOVER;
-- Expected: Threshold 30 seconds, Target orcls, Observer not yet
registered.

```

Schritt 4: Installieren Sie Oracle Instant Client auf dem Observer-Host

```

# On oradg-obs, as root (use -el8 / -el9 if the Observer is on an older
OL/RHEL)
sudo dnf install -y oracle-instantclient-release-el10
sudo dnf install -y oracle-instantclient-basic \
                oracle-instantclient-sqlplus \
                oracle-instantclient-tools

# Dedicated 'oracle' OS user (owns the wallet and the systemd unit)
sudo useradd -u 54321 -m oracle
sudo passwd -l oracle

# Oracle environment for the user
sudo mkdir -p /etc/oracle/network/admin
sudo chown -R oracle:oracle /etc/oracle
sudo -u oracle tee /home/oracle/.bash_profile >/dev/null <<'EOF'
export ORACLE_HOME=/usr/lib/oracle/26/client64
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export PATH=$ORACLE_HOME/bin:$PATH
export TNS_ADMIN=/etc/oracle/network/admin
EOF

# tnsnames.ora - must reach both DB hosts on TCP/1521
sudo tee /etc/oracle/network/admin/tnsnames.ora >/dev/null <<'EOF'
orcl =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = oracdb1) (PORT = 1521))
      (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME =
orcl)))
orcls =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = oracdb2) (PORT = 1521))
      (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME =
orcls)))
EOF
sudo chown oracle:oracle /etc/oracle/network/admin/tnsnames.ora

```

Schritt 5: Starten Sie den Observer als systemd-Einheit

Die Anmeldeinformationen werden in einer Auto-Login-Wallet gespeichert – niemals auf einer `dgmgr1` Befehlszeile (sichtbar für `ps/ journalctl`). Verbinden Sie sich `/@<tns_alias>` nur auf dem Observer.



- **Verwenden Sie ein dediziertes Konto:** Speichern Sie die Anmeldeinformationen für ein dediziertes Data Guard-Administratorenkonto (z. B. SYSDG) in der Wallet anstatt SYS.
- **Automatische Anmelde-Wallet erforderlich:** Der Observer-Systemd-Dienst benötigt eine automatische Anmelde-Wallet `cwallet.sso`). Wenn `cwallet.sso` nach der Ausführung von `mkstore` fehlt, verwenden Sie `orapki` aus dem Instant Client **tools**-Paket oder einem Datenbank-Home, um die automatische Anmelde-Wallet zu erstellen, und fügen Sie dann die gespeicherten Anmeldeinformationen erneut hinzu.

1. Erstellen Sie die Wallet auf dem Observer mit den Anmeldeinformationen beider Mitglieder:

```
sudo -iu oracle bash <<'BASH'
mkdir -p $TNS_ADMIN/wallet
mkstore -wrl $TNS_ADMIN/wallet -create      # prompts for a wallet password
- store in your secrets manager
mkstore -wrl $TNS_ADMIN/wallet -createCredential orcl sys ChangeMe!1
mkstore -wrl $TNS_ADMIN/wallet -createCredential orcls sys ChangeMe!1
BASH

sudo tee /etc/oracle/network/admin/sqlnet.ora >/dev/null <<'EOF'
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY =
/etc/oracle/network/admin/wallet)))
SQLNET.WALLET_OVERRIDE = TRUE
EOF
sudo chown oracle:oracle /etc/oracle/network/admin/sqlnet.ora
sudo chmod -R 0700 /etc/oracle/network/admin/wallet

sudo -iu oracle ls -l /etc/oracle/network/admin/wallet
# Expected: cwallet.sso and ewallet.p12

sudo -iu oracle bash <<'BASH'
sqlplus -L "/@orcl as sysdba" <<'SQL'
SELECT database_role FROM v$database;
EXIT
SQL
BASH

sudo -iu oracle bash <<'BASH'
sqlplus -L "/@orcls as sysdba" <<'SQL'
SELECT database_role FROM v$database;
EXIT
SQL
BASH

sudo -iu oracle dgmgrrl /@orcl 'SHOW CONFIGURATION;'
sudo -iu oracle dgmgrrl /@orcls 'SHOW CONFIGURATION;'
```

```
sudo -iu oracle orapki wallet create \  
-wallet /etc/oracle/network/admin/wallet \  
-auto_login  
sudo -iu oracle ls -l /etc/oracle/network/admin/wallet  
# Expected: cwallet.sso and ewallet.p12
```

1. Systemd-Unit + Logrotation:

```
sudo tee /etc/systemd/system/dgmgml-observer.service >/dev/null <<'EOF'  
[Unit]  
Description=Oracle Data Guard Fast-Start Failover Observer  
After=network-online.target  
Wants=network-online.target  
  
[Service]  
Type=simple  
User=oracle  
Group=oracle  
Environment=ORACLE_HOME=/usr/lib/oracle/26/client64  
Environment=LD_LIBRARY_PATH=/usr/lib/oracle/26/client64/lib  
Environment=TNS_ADMIN=/etc/oracle/network/admin  
Environment=PATH=/usr/lib/oracle/26/client64/bin:/usr/bin:/bin  
ExecStart=/usr/lib/oracle/26/client64/bin/dgmgml -silent /@orcl "START  
OBSERVER FILE IS '/var/lib/oracle/dgmgml-observer.dat'"  
Restart=always  
RestartSec=5  
  
[Install]  
WantedBy=multi-user.target  
EOF
```

```

sudo install -d -o oracle -g oracle -m 0755 /var/lib/oracle
sudo install -o oracle -g oracle -m 0640 /dev/null /var/log/dgmgml-
observer.log

sudo tee /etc/logrotate.d/dgmgml-observer >/dev/null <<'EOF'
/var/log/dgmgml-observer.log {
    weekly
    rotate 8
    compress delaycompress missingok notifempty
    create 0640 oracle oracle
    copytruncate
}
EOF

sudo systemctl daemon-reload && sudo systemctl enable --now dgmgml-
observer.service
sudo systemctl status dgmgml-observer.service

```

Überprüfung anhand der Primärdaten – Beobachter muss CONNECTED lesen (ein DISCONNECTED Beobachter setzt FSFO stillschweigend aus):

```

DGMGRL> SHOW FAST_START FAILOVER;
DGMGRL> SHOW CONFIGURATION;          -- Configuration Status: SUCCESS, FSFO:
ENABLED

```

Schritt 6: FSFO testen (Umschaltung und Failover)

Geplante Umstellung:

```

DGMGRL> VALIDATE DATABASE 'orcls';
DGMGRL> SWITCHOVER TO 'orcls';
DGMGRL> SHOW CONFIGURATION;
DGMGRL> SWITCHOVER TO 'orcl';      -- restore topology

```

Ungeplantes Failover: Führen Sie in einem Testfenster einen VM-Reset (Absturztest) durch; ein normaler **Stopp** löst möglicherweise kein FSFO aus. Tail /var/log/dgmgml-observer.log on oradg-obs; stellen Sie die Topologie nach Abschluss wieder her.

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFT SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.